

Counter

The Counter is a digital sequential circuit that consists of group of flipflop connected in cascade whose function is to count the number of input pulses. A counter is a sequential machine that produces a specified count sequence. The count changes whenever the input clock is asserted.

A Counter can be constructed by a synchronous circuit or by an asynchronous circuit. With a synchronous circuit, all the bits in the count change synchronously with the assertion of clock.

With an asynchronous circuit, all the bits in the count do not change at the same time.

In an asynchronous (ripple) counter, flipflop output transition serves as a source for triggering other flipflop. In other words, the clock pulse (CP) input of all flipflop except first are triggered not by incoming pulses, but rather by the transition that occurs in other flipflops.

In synchronous counter, the input pulses are applied to all clock pulse (CP) input of all flipflop.

A_nsynchronous \leftrightarrow Ripple

292

Asynchronous counter vs. Synchronous counter

Asynchronous counter

- 1) Flip Flop are connected in such a way that the output of first flip flop drives the clock of next flip flop.

- 2) Flip Flop are not clocked simultaneously.

- 3) Less number of logic gates are required.

- 4) These are slow in operation.

- 5) The cost is low.

- 6) Speed is slow as clock is propagated through number of stages.

Synchronous counter

- 1) There is no connection between output of the first flip flop and clock of next flip flop.

- 2) All flip flops are clocked simultaneously.

- 3) Large number of logic gates are required to design.

- 4) These are faster than that of Ripple counter.

- 5) The cost is high.

- 6) Speed is high as clock is given at a same time.

Counters are classified into two categories:-

- 1) Asynchronous counters (Ripple counters)
- 2) Synchronous counters

1) Asynchronous Counter

An asynchronous counter is one in which the counter do not change states at exactly the same time because they do not have a common clock pulse.

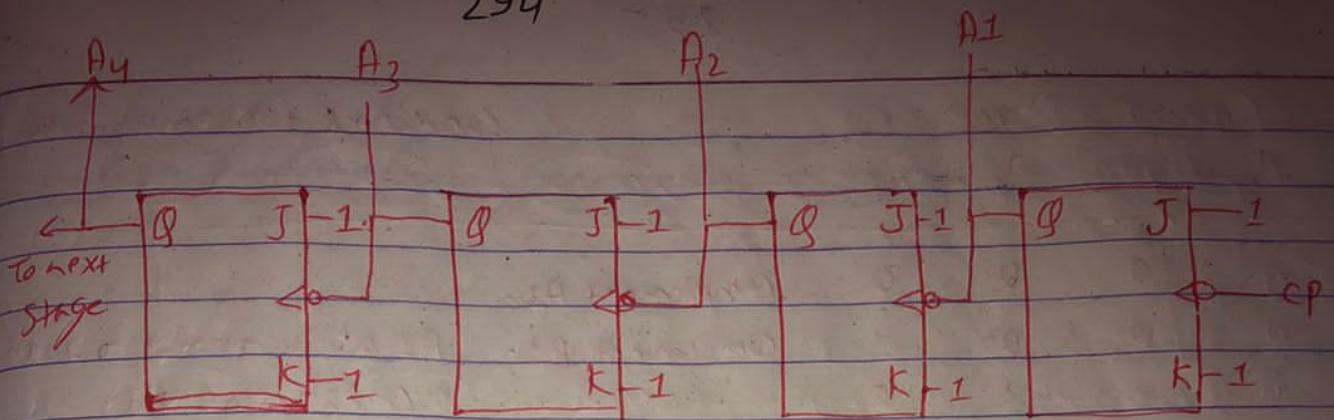
The main characteristic of an asynchronous counter is each flip flop derives its own clock from other flip flops and is therefore independent of the input clock.

a) Binary Ripple Counter

A binary ripple counter consists of a series connection of complementing flip flops (T or JK type) with the output of each flip flop connected to the CP input of the next higher-order flip flop. The flip flop holding the least significant bit (LSB) receives the incoming count pulses.

The diagram of a 4-bit Binary ripple counter is shown in fig(a).

All J and K inputs are equal to 1.



fig(a): 4-bit Binary ripple counter

The small circle in the CP input indicated that the flipflop complements during a negative-going transition or when the output to which it is connected goes from 1 to 0.

To understand the operation of the binary counter, refer to its count sequence given in Table -

It is obvious that the lowest-order bit A1 must be complemented with each count pulse. Every time A1 goes from 1 to 0, it complements A2. Every time A2 goes from 1 to 0, it complements A3 and so on.

Count Sequence

A₄ A₃ A₂ A₁

Conditions for complementing F

0 0 0 0 complement A₁.

0 0 0 1 complement A₁. A₁ will go from 1 to 0 and complement A₂.

0 0 1 0

0 0 1 1 complement A₁.

0 1 ← 0 ← 8 complement A₁. A₂ will go from 1 to 0 and complement A₂; A₂ will go from 1 to 0 and complement A₃.

0 1 0 1 complement A₁.

0 1 1 0 complement A₁. A₁ will go from 1 to 0 and complement A₂.

0 1 1 1 complement A₁.

1 ← 0 ← 0 ← 0 ← 0 complement A₁. A₁ will go from 1 to 0 and complement A₂; A₂ will go from 1 to 0 and complement A₃; A₃ will go from 1 to 0 and complement A₄.

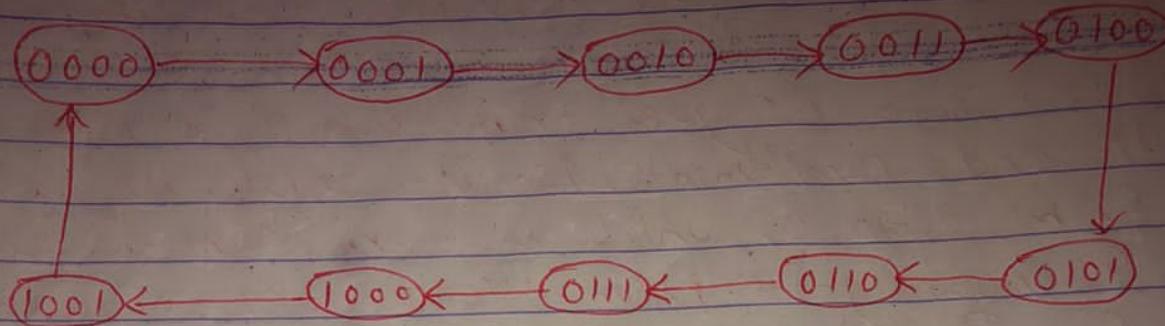
So on - - - .

For example: take the transition from count 0111 to 1000. A1 is complemented with the count pulse. Since A1 goes from 1 to 0, it triggers A2 and complements it. As a result, A2 goes from 1 to 0, which in turn complements A3. A3 now goes from 1 to 0, which complements A4. The output transition of A4, if connected to a next stage, will not trigger the next flip-flop since it goes from 0 to 1. The flip-flop change one at a time in rapid succession, and the signal propagates through the counter in a ripple fashion.

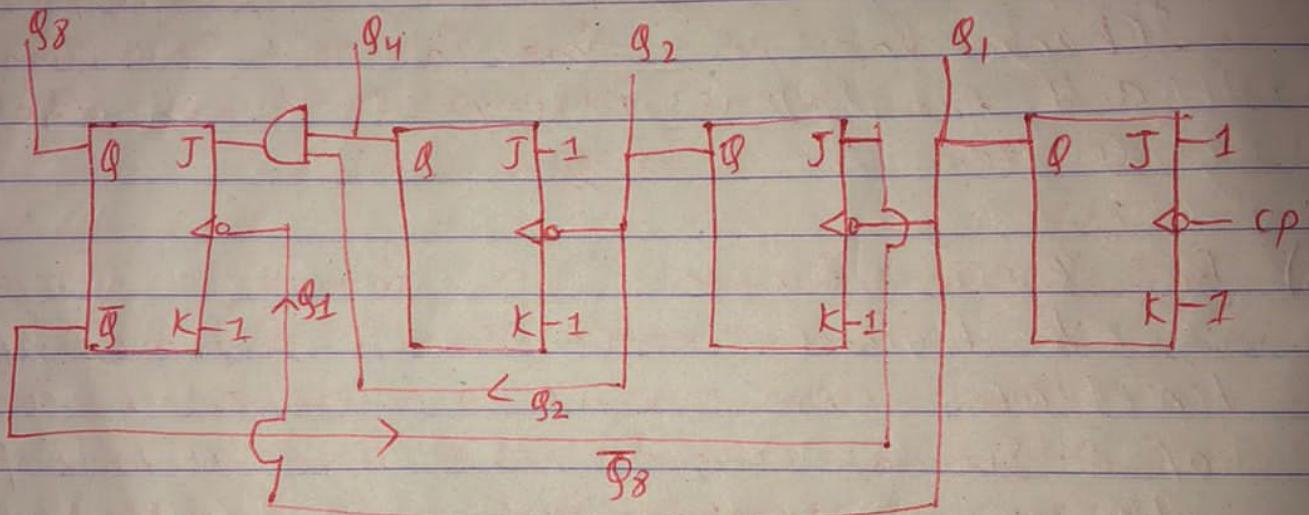
b) BCD Ripple Counter (Decade Counter)

A decimal counter follows a sequence of ten states and returns to 0 after the count of 9. Such a counter must have at least four flip-flops to represent each decimal digit, since a decimal digit is represented by a binary code with at least four bits. The sequence of states in a decimal counter is dictated by the binary code used to represent a decimal digit.

If BCD is used, the sequence of states is as shown in the state diagram. This is similar to a binary counter, except that the state after 1001 (9) is 0000 (0).



fig(a): State diagram of BCD counter.



fig(b): BCD ripple counter

The four outputs are designated by the letter symbol Q with a numeric subscript equal to the binary weight of the corresponding bit in the BCD code.

The flipflop triggers on the negative edge.

Operation

When CP input goes from 1 to 0, the flip-flop is set if $J=1$, is cleared if $K=1$, is complemented if $J=K=1$, and is left unchanged if $J=K=0$.

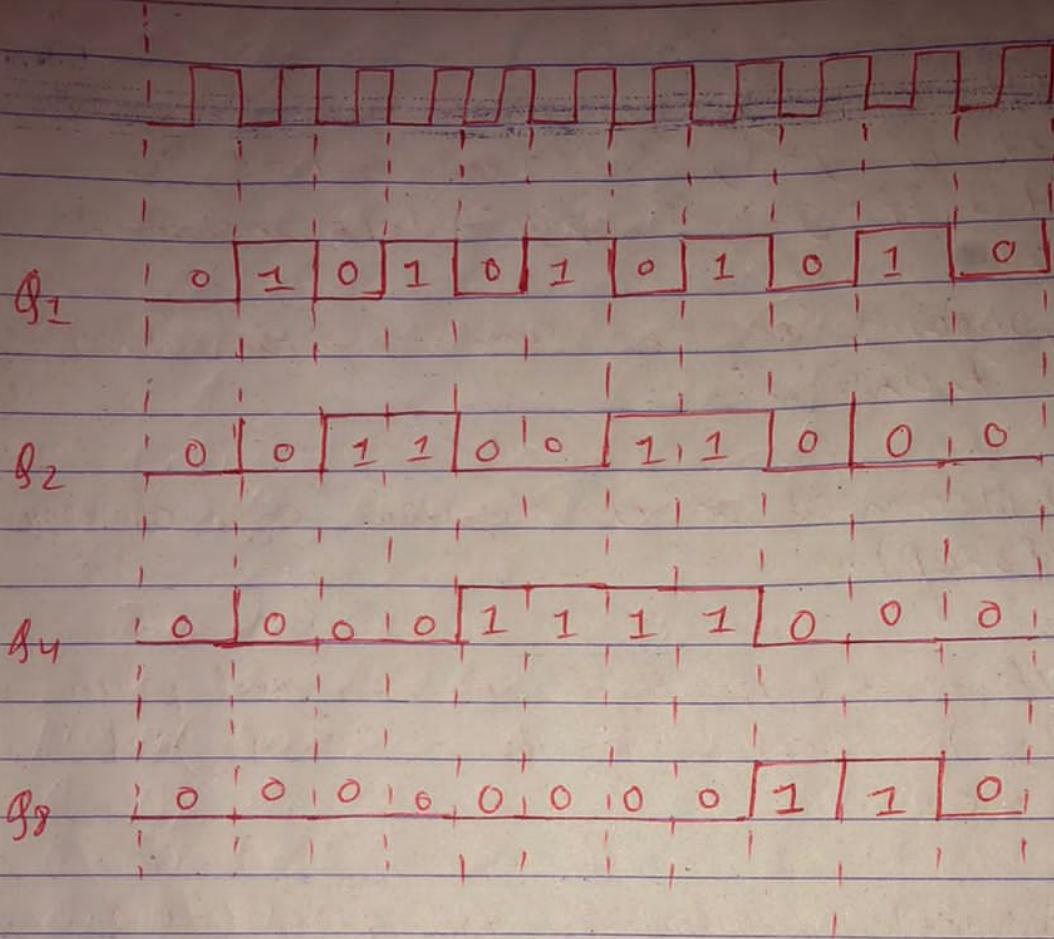
The following are the conditions for each flip-flop state transition:

i) Q_1 is complemented on the negative edge of every count pulse (CP).

ii) Q_2 is complemented if $Q_3=0$ and Q_1 goes from 1 to 0. Q_2 is cleared if $Q_3=1$ and Q_1 goes from 1 to 0.

iii) Q_4 is complemented when Q_2 goes from 1 to 0.

iv) Q_3 is complemented when $Q_4 Q_2 = 11$ and Q_1 goes from 1 to 0. Q_3 is cleared if either Q_4 or Q_2 is 0 and Q_1 goes from 1 to 0.



fig(c): Timing diagram of BCD counter.



7) Synchronous counter

In Synchronous counter, the clock inputs of all the flip flops are connected together and are triggered by the input pulse. Thus, all the flip flops change state simultaneously (in parallel).

a) 3-bit Synchronous Binary up counter

An N -bit Synchronous counter binary up counter consists of ' N ' T-Flipflops. It counts from 0 to $2^N - 1$.

The 3-bit synchronous binary up counter contains 3 T-flipflops and one 2-input AND gate. All these flipflops are negative edge triggered and the outputs of flipflops change synchronously. The T-input of the first, second and third flipflops are T , Q_A and $Q_A Q_B$ respectively. (After using K-map).

The output of first T-Flipflop toggles for every negative edge of a clock signal. The output of second T-flipflop toggles for every negative edge of clock signal if Q_A is 1. The output of third T-flipflop toggles for every negative edge of a clock signal if both Q_A and Q_B are 1.

3a1

The state diagram for 3-bit synchronous binary up counter is given as:-

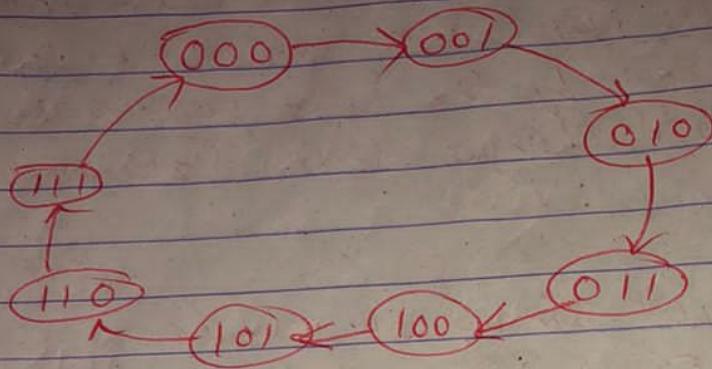


fig: State diagram

The truth table is given as:-

present state	Next State	Input to Flipflop		
q_C q_B q_A	q_C^+ q_B^+ q_A^+	T_C	T_B	T_A
0 0 0	0 0 1	0	0	1
0 0 1	0 1 0	0	1	1
0 1 0	0 1 1	0	0	1
0 1 1	1 0 0	1	1	1
1 0 0	1 0 1	0	0	1
1 0 1	1 1 0	0	1	1
1 1 0	1 1 1	0	0	1
1 1 1	0 0 0	1	1	1

From table, we get $T_A = 1$.

302

For T_B , using k-map, we have

$Q_B Q_A$	$Q_B Q_A'$	$Q_B' Q_A$	$Q_B' Q_A'$
Q_C	0	1	1
Q_C'	0	1	1
Q_C	0	1	1

$$\boxed{T_B = Q_B Q_A}$$

For T_C ,

$Q_B Q_A$	$Q_B Q_A'$	$Q_B' Q_A$	$Q_B' Q_A'$
Q_C	0	1	1
Q_C'	0	1	1
Q_C	0	1	1

$$\boxed{T_C = Q_B Q_A}$$

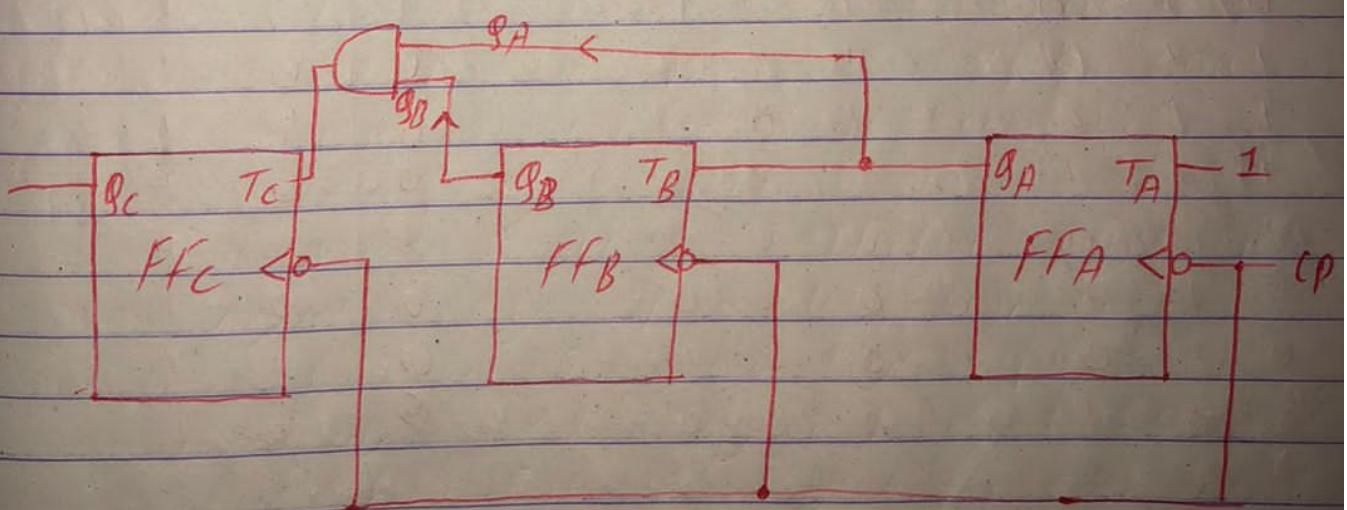


fig: 3 bit synchronous binary up counter

b) 3-bit Synchronous binary down counter

The state diagram for 3-bit synchronous binary down counter is given below:-

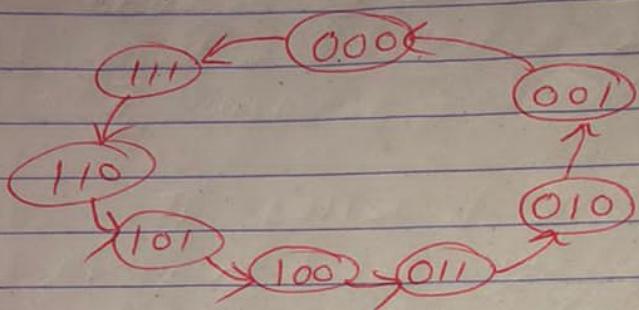


fig: State diagram of 3-bit down counter.

The truth table for 3-bit synchronous binary down counter using T-Flipflop is drawn as:-

Present State	Next State	Input to Flipflop
$Q_2 Q_1 Q_0$	$Q_2^+ Q_1^+ Q_0^+$	$T_2 \quad T_1 \quad T_0$
0 0 0	1 1 1	1 1 1
1 1 1	1 1 0	0 0 1
1 1 0	1 0 1	0 1 1
1 0 1	1 0 0	0 0 1
1 0 0	0 1 1	1 1 1
0 1 1	0 1 0	0 0 1
0 1 0	0 0 1	0 1 1
0 0 1	0 0 0	0 0 1

From table, we get $T_0 = 1$.

304

For T_1 ,

$Q_1 Q_0$	$Q'_1 Q'_0$	$Q'_1 Q_0$	$Q_1 Q_0$	$Q'_1 Q'_0$
Q_2	1	0	0	1
Q'_2	1	0	0	1
Q_2	1	0	0	1

for T_2

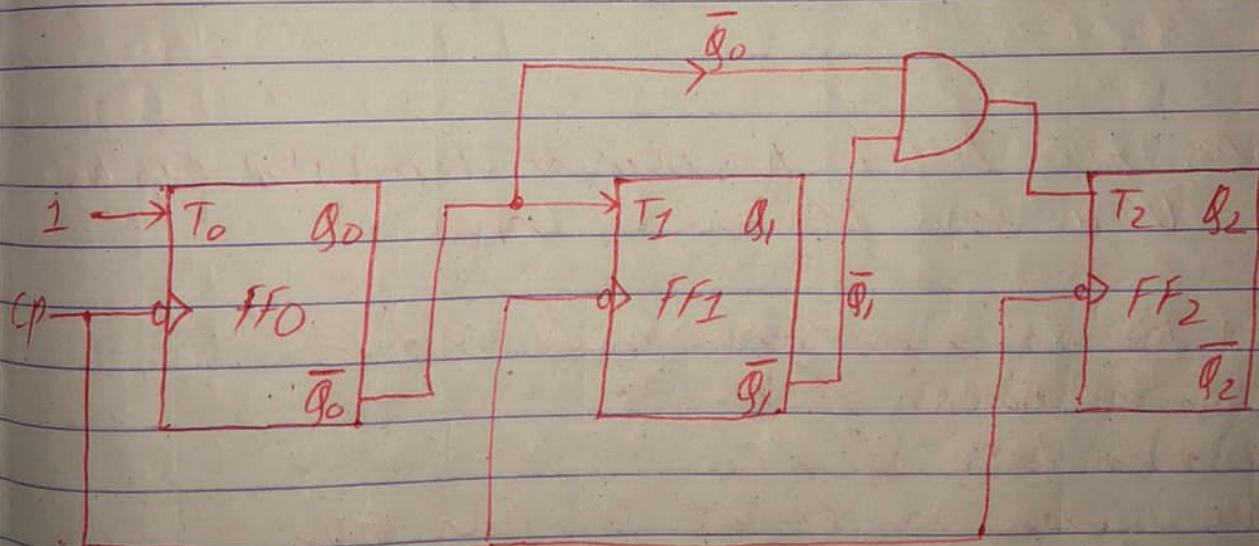
$Q_1 Q_0$	$Q'_1 Q'_0$	$Q'_1 Q_0$	$Q_1 Q_0$	$Q'_1 Q'_0$
Q_2	1	0	0	0
Q'_2	1	0	0	0
Q_2	1	0	0	0

$$\begin{aligned} T_1 &= Q'_1 Q'_0 + Q_1 Q'_0 \\ &= Q'_0 (Q'_1 + Q_1) \\ &= Q'_0 \cdot 1 \end{aligned}$$

$$T_1 = Q'_0$$

$$T_2 = Q'_1 Q'_0$$

The block diagram of a 3-bit synchronous binary down counter is shown in figure below:-



An N -bit synchronous binary down counter consists of ' N ' T-flipflops. It counts from 2^{N-1} to 0.

The 3-bit synchronous binary down counter consists of 3 T-flipflops and one 2-input AND gate. All these flipflops are negative edge triggered and the output of flipflop change synchronously. The T-inputs of first, second and third flipflop are T , \bar{q}_0 and $\bar{q}_0 \bar{q}_1$, respectively.

The output of first T-FF toggles for every negative edge of a clock signal. The output of second T-FF toggles for every negative edge of clock signal if \bar{q}_0 is 1. The output of third T-FF toggles for every negative edge of clock signal if both \bar{q}_0 and \bar{q}_1 are 1.

Note: For N-bit, N-number of flipflop can be used and proceed as before.

Steps for designing synchronous counter.

- 1) Decide the number of FF.
- 2) Excitation table of FF.
- 3) State diagram and circuit excitation table
- 4) Obtain simplified equations using k-map.
- 5) Draw the logic/block diagram.

1) Synchronous Binary Up-down Counter (Bidirectional Counter)

Bidirectional counters also known as up/down counter are capable of counting in either direction through any given count sequence and they can be reversed at any point within their count sequence by using additional control input.

The state diagram for the bidirectional counter is given as:-

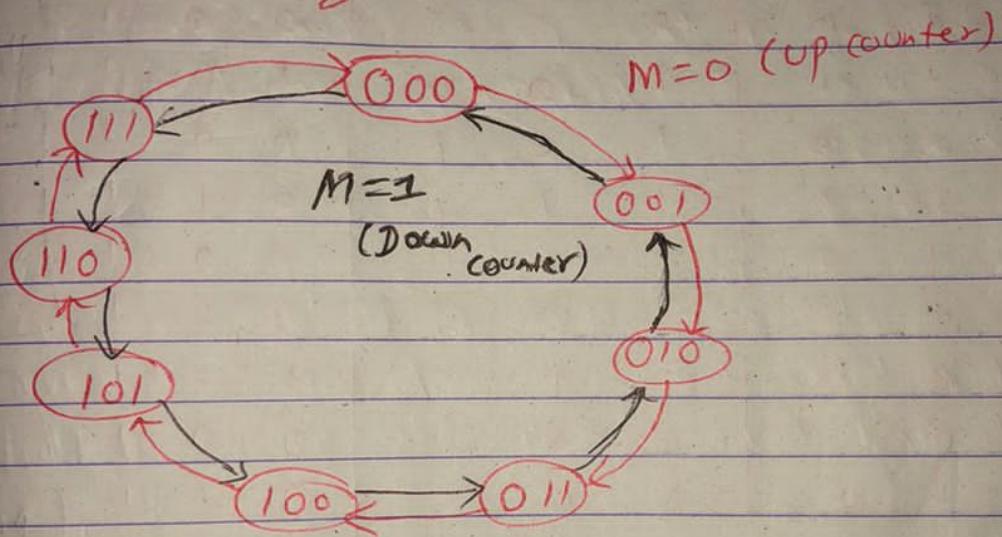


Fig: State diagram of bidirectional counter.

The excitation table for T-FF is given as:-

S	S^+	T
0	0	0
0	1	1
1	0	1
1	1	0

The state table for the bidirectional counter
P8 illustrated in the table:

Control Input	Present State			Next State			Input to FF		
M	Q_C	Q_B	Q_A	Q_C^+	Q_B^+	Q_A^+	T_C	T_B	T_A
0	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	1	0	0	1
0	0	1	1	1	0	0	1	1	1
0	1	0	0	1	0	1	0	0	1
0	1	0	1	1	1	0	0	1	1
0	1	1	0	1	1	1	0	0	1
0	1	1	1	0	0	0	1	1	1
1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	0	0	0	1
1	0	1	0	0	0	1	0	1	1
1	0	1	1	0	1	0	0	0	1
1	1	0	0	0	1	1	1	1	1
1	1	0	1	1	0	0	0	0	1
1	1	1	0	1	0	1	0	1	1
1	1	1	1	1	1	0	0	0	1

UP Counting

Down Counting

From state table, we get

$$T_A = 1.$$

for T_C

$M'Q_A Q'_A$	$Q'_B Q_A$	$Q'_B Q'_A$	$Q_B Q_A$	$Q_B Q'_A$
$M'Q_C$	0 0	0 1	1 3	0 2
$M'Q'_C$	0 4	0 5	1 7	0 6
$M''Q_C$	0 12	0 13	0 15	0 14
$M''Q'_C$	1 8	0 9	0 11	0 10

for T_B

$M'Q_A Q'_A$	$Q'_B Q_A$	$Q'_B Q'_A$	$Q_B Q_A$	$Q_B Q'_A$
$M'Q_C$	0	1	1	0
$M'Q'_C$	0	1	1	0
$M''Q_C$	1	0	0	1
$M''Q'_C$	1	0	0	1

$$T_C = M'Q'_B Q_A + M'Q_B Q'_A$$

$$T_B = M'Q_A + M'Q'_A$$

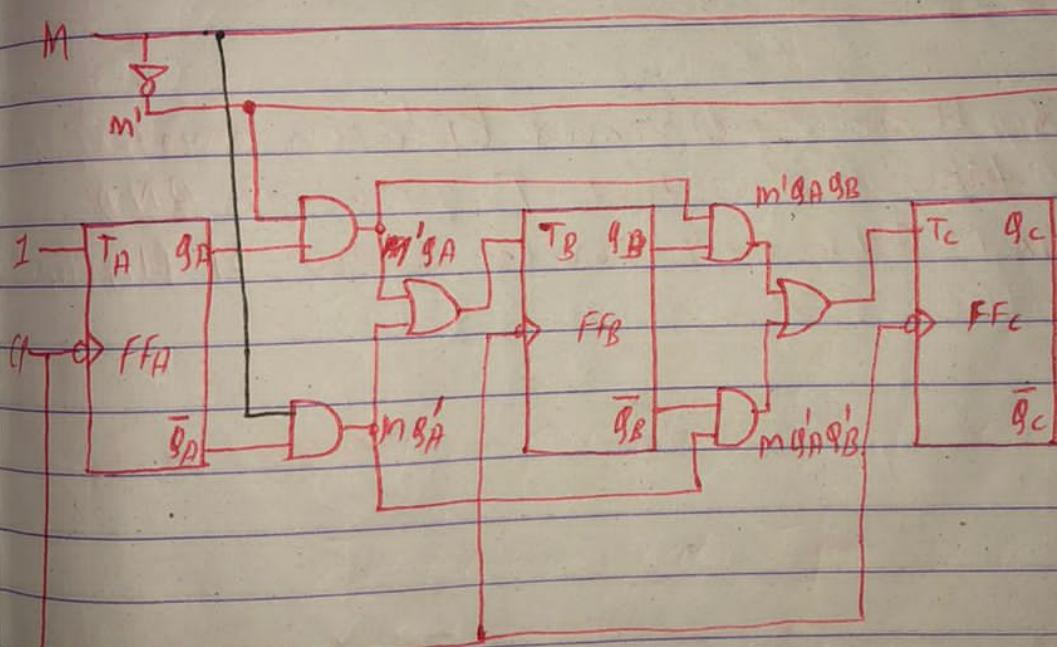


fig: binary ^{3-bit} up/down counter

309

When $M=0$, the circuit counts up.
When $M=1$, the circuit counts down.

The 3-bit synchronous binary up down counter consists of 3-T-flipflops, four 2-input AND gate and two 2-input OR gate. All these flipflops are negative edge triggered and the output of flipflop change synchronously. The T-input of the first, second and third flipflop are 1, $M'Q_A + Q_A'$ and $M'Q_B Q_B' + M Q_B'$ respectively.

Note: For a 4-bit synchronous up/down counter,
simply take the output of previous AND gate and AND with current output of flipflop.

(a) Design a Synchronous Mod-6 counter using Clocked T-flipflop.

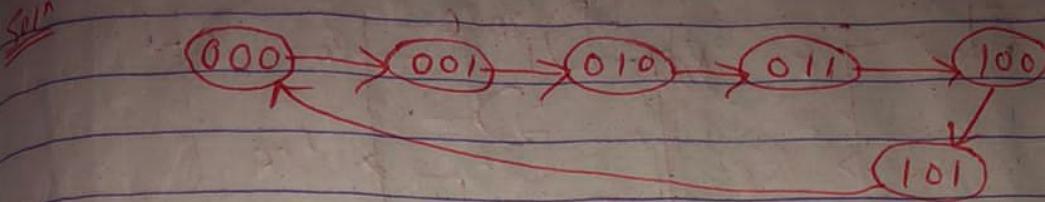


fig: State diagram of Mod-6 Counter

Count	Q_C present state	Q_B	Q_A	Q_C^+ Next state	Q_B^+	Q_A^+	T_E Flip Flop	T_B Input	T_A Input
0	0	0	0	0 0 1			0	0	1
1	0	0	1	0 1 0			0	1	1
2	0	1	0	0 1 1			0	0	1
3	0	1	1	1 0 0			1	1	1
4	1	0	0	1 0 1			0	0	1
5	1	0	1	0 0 0	-	-	1	0	1
6	1	1	0	X X X			X	X	X
7	1	1	1	X X X			X	X	X

From state-table, $T_A = 1$.

For T_B

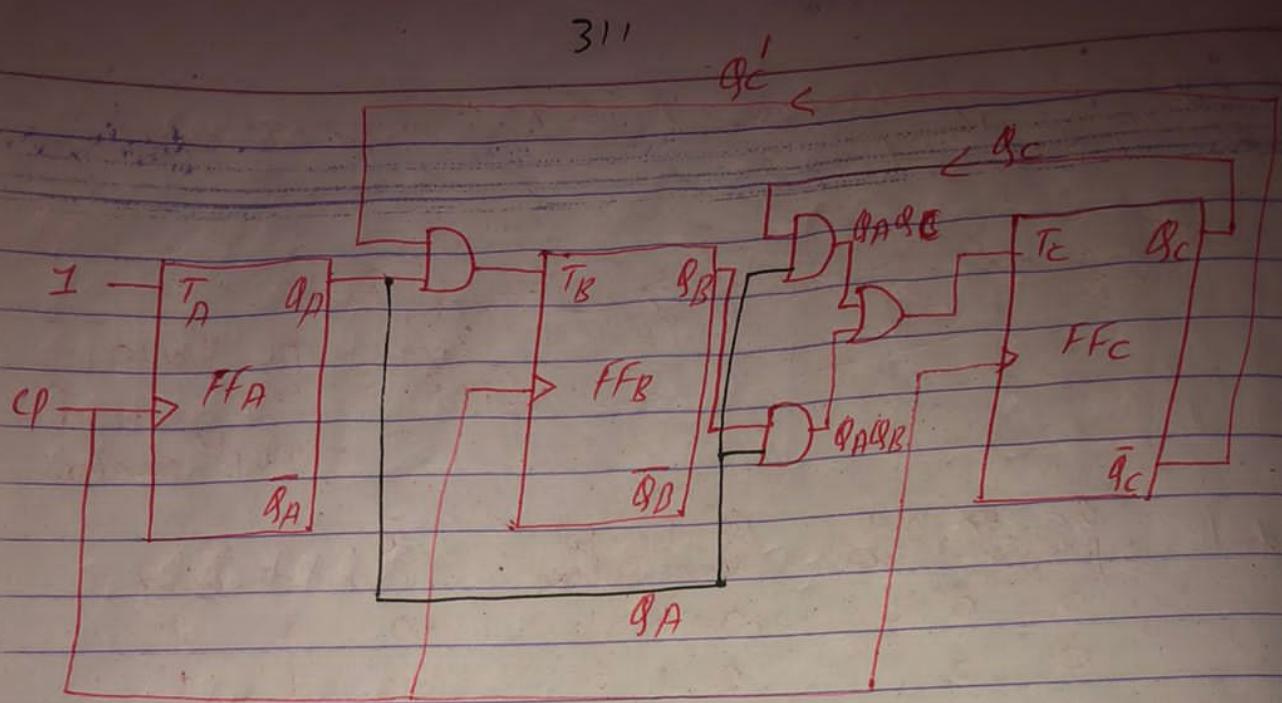
Q_C	$Q_B Q_A$	$Q_B' Q_A$	$Q_B Q_A'$	$Q_B' Q_A'$
Q'_C	0 1 D 0			
Q_C	0 0 X X			

$$\therefore T_B = Q'_C Q_A$$

For T_C

Q_C	$Q_B Q_A$	$Q_B' Q_A$	$Q_B Q_A'$	$Q_B' Q_A'$
Q'_C	0 0 1 0			
Q_C	0 0 1 X			
Q'_C	0 1 X X			

$$T_C = Q_C Q_A + Q'_B Q_A$$



Q) Design a synchronous Mod-10 counter using T-FF.
(BCD counter)

Soln

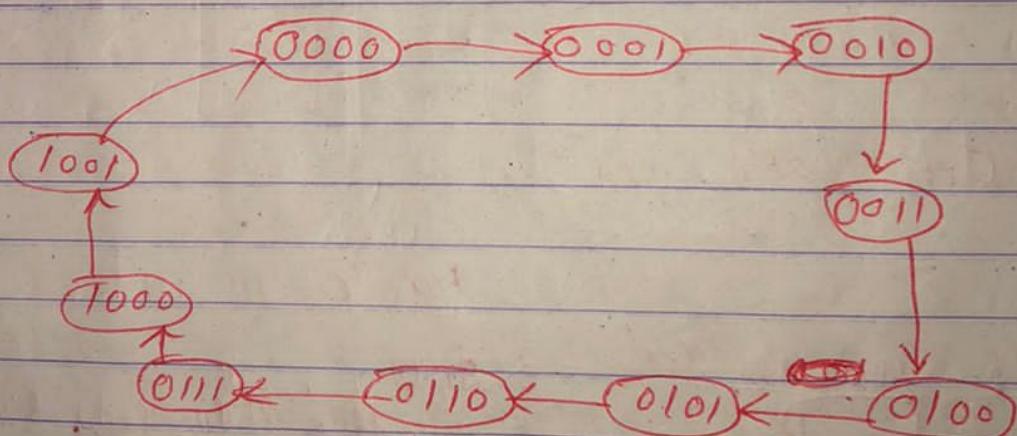


fig: State diagram of BCD counter/
Mod-10 counter

The State table for Mod-10 counter is given as:-

ILK	present State	Next State	Flip flop	Input
	$Q_3\ Q_2\ Q_1\ Q_0$	$Q_3^+\ Q_2^+\ Q_1^+\ Q_0^+$	$T_3\ T_2\ T_1\ T_0$	
0	0 0 0 0	0 0 0 1	0 0 0 1	1
1	0 0 0 1	0 0 1 0	0 0 0 1	1
2	0 0 1 0	0 0 1 1	0 0 0 0	1
3	0 0 1 1	0 1 0 0	0 1 1 1	1
4	0 1 0 0	0 1 0 1	0 0 0 0	1
5	0 1 0 1	0 1 1 0	0 0 0 1	1
6	0 1 1 0	0 1 1 1	0 0 0 0	1
7	0 1 1 1	1 0 0 0	1 1 1 1	1
8	1 0 0 0	1 0 0 1	0 0 0 0	1
9	1 0 0 1	0 0 0 0	1 0 0 1	1
10	1 0 1 0	X X X X	X X X X	X
11	1 0 1 1	X X X X	X X X X	X
12	1 1 0 0	X X X X	X X X X	X
13	1 1 0 1	X X X X	X X X X	X
14	1 1 1 0	X X X X	X X X X	X
15	1 1 1 1	X X X X	X X X X	X

From state table, $T_0 = 1$.

313

For T_3 ,

$Q_3 Q_2$	$Q_3 Q_0$	$Q'_3 Q_0$	$Q'_3 Q_2$	$Q_1 Q_0$	$Q_1 Q_2$	$Q'_1 Q_0$	$Q'_1 Q_2$
$Q_3 Q_2$	0^0	0^1	0^3	0^2			
$Q'_3 Q_2$	0^4	0^5	1^2	0^6			
$Q'_3 Q_2$	0^12	0^13	1^5	0^14			
$Q_3 Q_2$	X^X	X^X	X^X	X^X			
$Q_3 Q'_2$	0^8	1^9	X^X	X^{10}			

For T_2

$Q_3 Q_2$	$Q_3 Q_0$	$Q'_3 Q_0$	$Q_1 Q_0$	$Q_1 Q_2$	$Q'_1 Q_0$	$Q'_1 Q_2$
$Q_3 Q_2$	0^0	0^1	1^3	0^2		
$Q'_3 Q_2$	0^4	0^5	1^7	0^6		
$Q'_3 Q_2$	X^{12}	X^{13}	X^{15}	X^{14}		
$Q_3 Q_2$	X^8	X^9	X^{11}	X^{10}		
$Q_3 Q'_2$	0^8	0^9	X^{11}	X^{10}		

$$T_3 = Q_3 Q_0 + Q_1 Q_1 Q_0$$

$$T_2 = Q_1 Q_0$$

For T_1 ,

$Q_3 Q_2$	$Q_3 Q_0$	$Q'_3 Q_0$	$Q_1 Q_0$	$Q_1 Q_2$	$Q'_1 Q_0$
$Q_3 Q_2$	0^0	1^1	1^3	0^2	
$Q'_3 Q_2$	0^4	1^5	1^7	0^6	
$Q'_3 Q_2$	X^{12}	X^{13}	X^{15}	X^{14}	
$Q_3 Q'_2$	0^8	0^9	X^{11}	X^{10}	

$$T_1 = Q'_3 Q_0$$

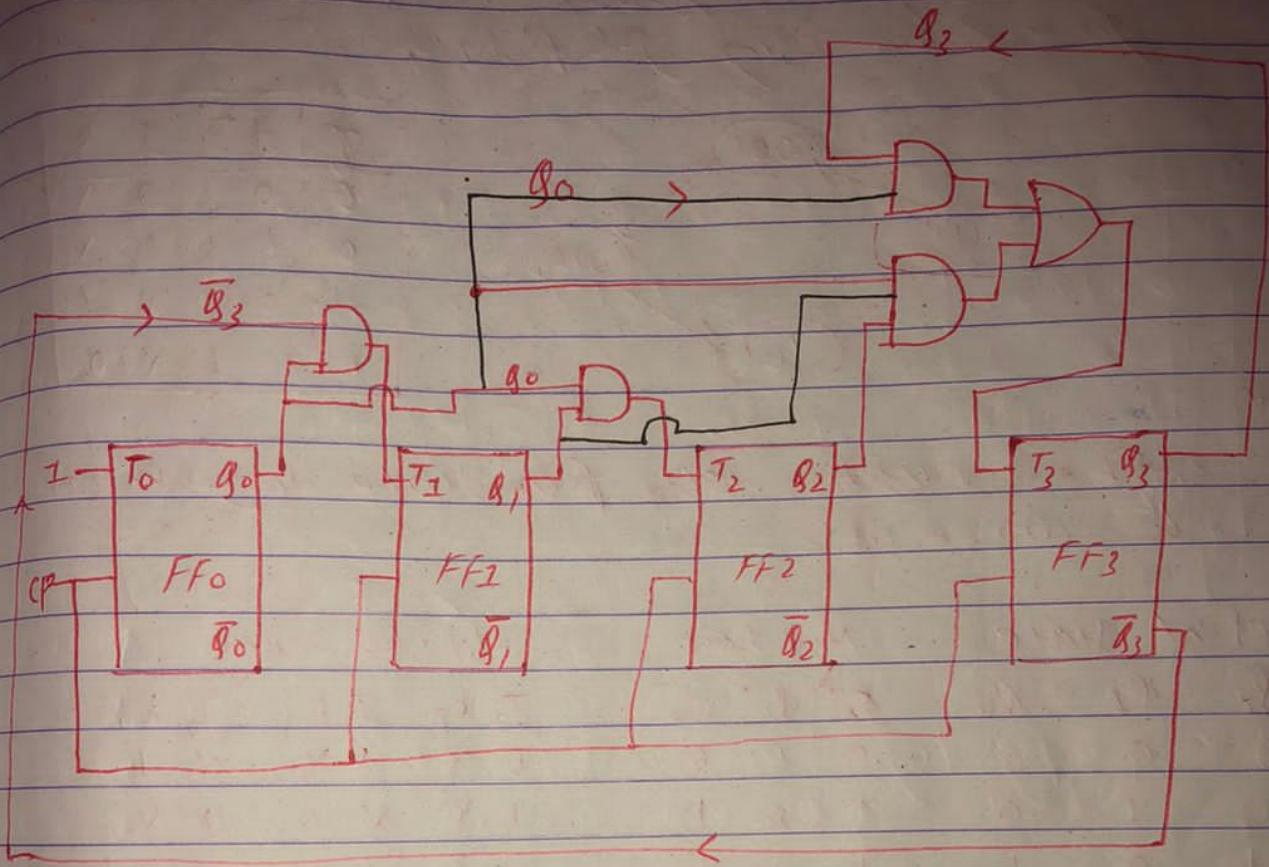
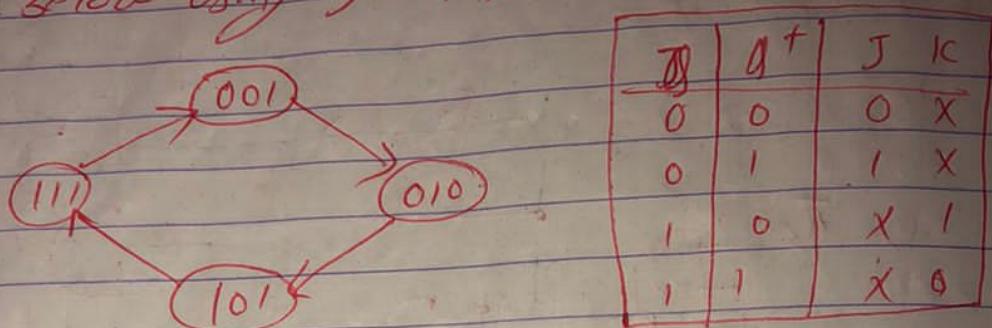


fig: Logic diagram of BCD Counter using
T- flipflop..

315

Q) Design a counter with the irregular binary count sequence shown in the state diagram given below using JK flip flop.

Soln

The transition table from state diagram can be drawn as:-

Present State			Next State			Flip Flop Input			
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	J_2	K_2	J_1, K_1	J_0, K_0
1	0	0	1	0	1	0	X	1	X
2	0	1	0	1	0	1	X	X	1
5	1	0	1	1	1	X	0	1	X
7	1	1	1	0	0	X	1	X	0

Don't care can be placed in the cells corresponding to the invalid states of 000, 011, 100, and 110 as marked by X's.

• 3 6

For J_2

q_2	$q'_1 q_0$	$q'_1 q_0'$	$q'_1 q_0$	$q_1 q_0$	$q_1 q'_0$
q'_2	x^0	0^1	$(x^3 \quad 1^2)$		
q_2	x^4	x^5	$(x^7 \quad x^6)$		

$$J_2 = q_1$$

For K_2

q_2	$q'_1 q_0$	$q'_1 q_0'$	$q'_1 q_0$	$q_1 q_0$	$q_1 q'_0$
q'_2	x^0	x^1	$(x^3 \quad x^2)$		
q_2	x^4	0^5	$(1^7 \quad x^6)$		

$$K_2 = q_1$$

For J_1

q_2	$q'_1 q_0$	$q'_1 q_0'$	$q'_1 q_0$	$q_1 q_0$	$q_1 q'_0$
q'_2	x^0	1^1	$(x^3 \quad x^2)$		
q_2	x^4	1^5	$(x^7 \quad x^6)$		

$$J_1 = I$$

For K_1

q_2	$q'_1 q_0$	$q'_1 q_0'$	$q'_1 q_0$	$q_1 q_0$	$q_1 q'_0$
q'_2	x^0	x^1	$(x^3 \quad 1^2)$		
q_2	x^4	x^5	$(1^7 \quad x^6)$		

$$K_1 = I$$

For J_0

q_2	$q'_1 q_0$	$q'_1 q_0'$	$q'_1 q_0$	$q_1 q_0$	$q_1 q'_0$
q'_2	x^0	x^1	$(x^3 \quad 1^2)$		
q_2	x^4	x^5	$(x^7 \quad x^6)$		

$$J_0 = I$$

For K_0

q_2	$q'_1 q_0$	$q'_1 q_0'$	$q'_1 q_0$	$q_1 q_0$	$q_1 q'_0$
q'_2	x^0	1^1	$(x^3 \quad x^2)$		
q_2	x^4	0^5	$(0^7 \quad x^6)$		

$$K_0 = q'_2$$

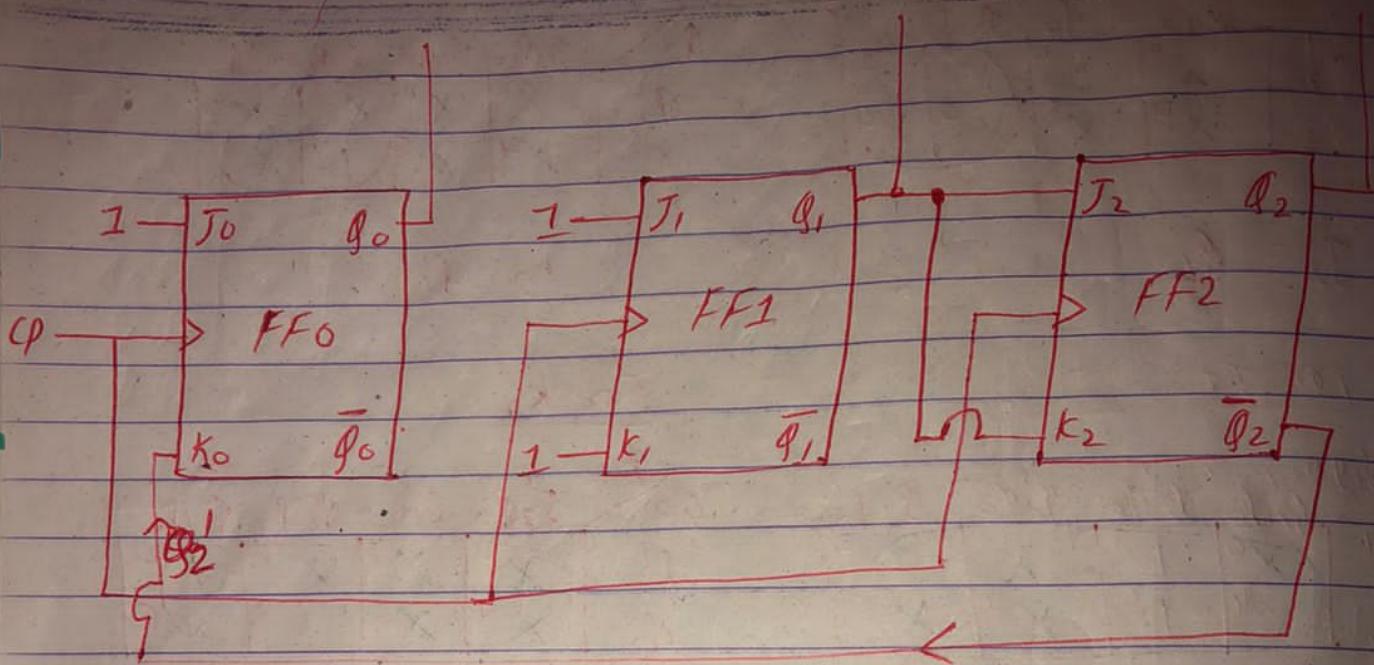
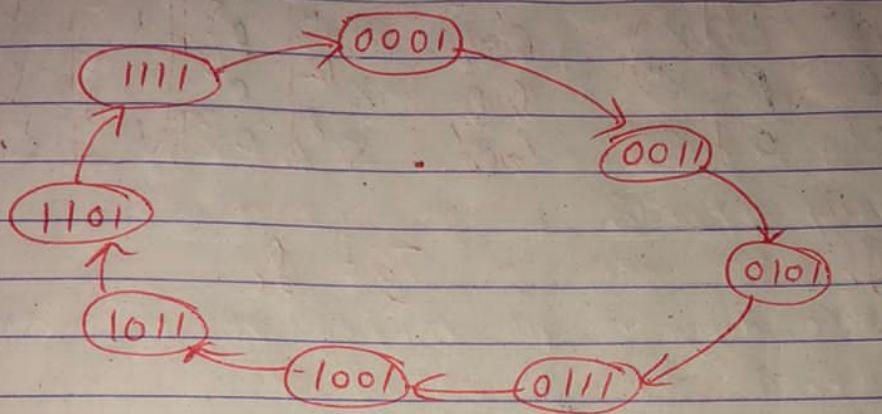


fig: Logic diagram of counter Using J-K flip flop.

Q) Design a synchronous counter which generates odd number from 0 to 15.

SOLN

The state diagram for this is drawn as



Present State	Next State	FlipFlop Input
$q_A\ q_B\ q_C\ q_D$	$q_A^+\ q_B^+\ q_C^+\ q_D^+$	$T_A\ T_B\ T_C\ T_D$

1	0 0 0 1	0 0 1 1	0 0 1 0
3	0 0 1 1	0 1 0 1	0 1 1 0
5	0 1 0 1	0 1 1 1	0 0 1 0
7	0 1 1 1	1 0 0 1	1 1 1 0
9	1 0 0 1	1 0 1 1	0 0 1 0
11	1 0 1 1	1 1 0 1	0 1 1 0
13	1 1 0 1	1 1 1 1	0 0 1 0
15	1 1 1 1	0 0 0 1	1 1 1 0

From State table, we have

$$\bar{T}_C = 1 \text{ and } \bar{T}_D = 0$$

For T_A

$q_A q_B$	$q'_A q'_B$	$q'_A q_B$	$q_A q_B$	$q_A q'_B$
X^0	0^1	0^3	X^2	
X^4	0^5	T^7	X^6	
X^{12}	0^{13}	1^5	X^{14}	
X^8	0^9	0^{11}	X^{10}	

 T_B

$q_A q_B$	$q'_A q'_B$	$q'_A q_B$	$q_A q_B$	$q_A q'_B$
X^0	0^1	1^3	X^2	
X^4	0^5	1^7	X^6	
X^{12}	0^{13}	1^{15}	X^{14}	
X^8	0^{9^+}	1^{11}	X^{10}	

$$T_A = \bar{q}_B q_C$$

$$T_B = q_C$$

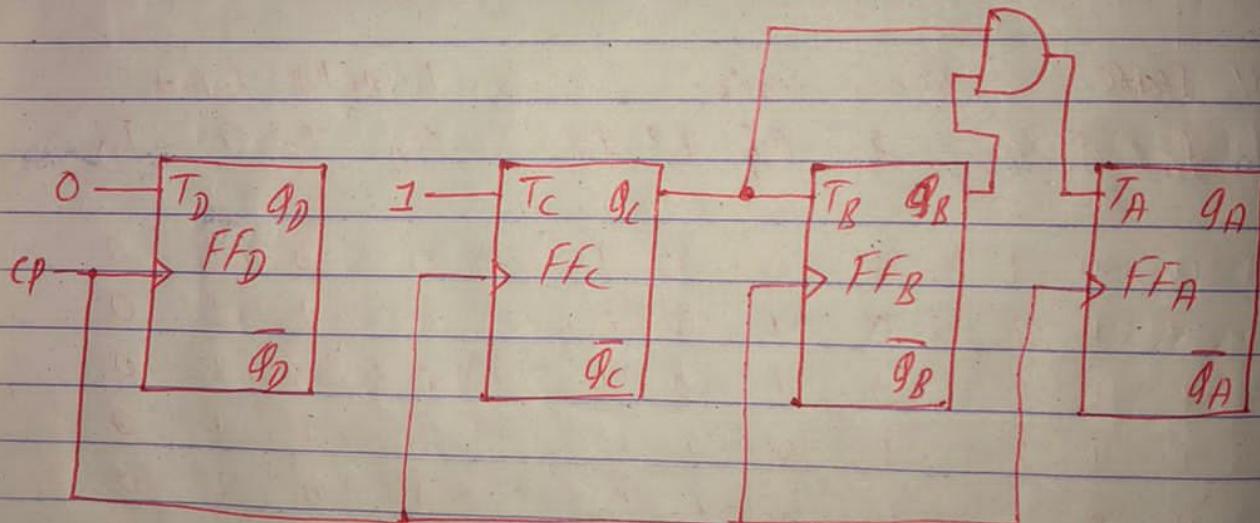


Fig: logic diagram of synchronous counter that generates odd number from 0 to 15.

Steps for Asynchronous Mod counter design

Step 1: Determine the number of states through which counter passes.

Step 2: Decide the number of bits or flipflops for ripple counter.

Step 3: Draw state diagram followed by state transition table for (Reset logic).

Step 4: Obtain the reset logic using K-map simplification.

Step 5: Draw the logic diagram.

Modulus of a counter (Mod counter)

Modulus of a counter indicates the number of states through which counter passes during its operation.

The number of flipflops required (m) to construct mod- n counter is $N \leq 2^m$.

For mod- n counter, the states will be n but the counting will be 1 CM i.e. for mod-6 counter, the counting will be from 0 to 5.

321

Q) Design a mod-5 ripple counter

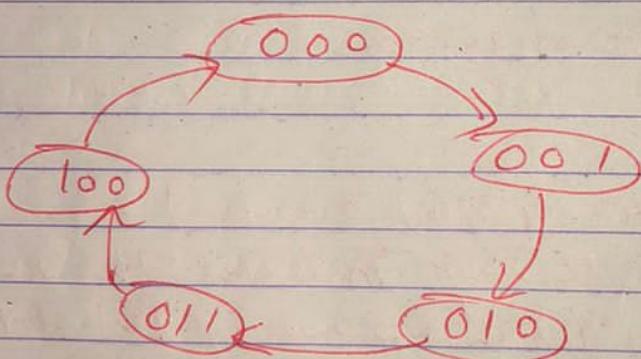
Sol

The number of states through which counter passes = 5

The states will go from 0 to 4.

To count 5 states, we need 3 bit ripple counter
No. of FF = 3

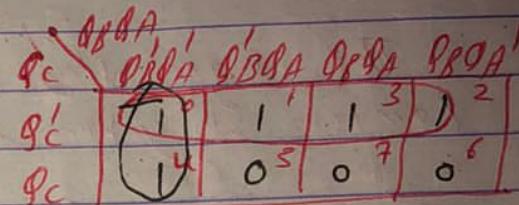
The state diagram for mod-5 ripple counter can be drawn as:-



The State transition table is given as:-

State	Q_C	Q_B	Q_A	Reset logic (Y)
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

States from 0 to 4 are only valid states.



$$Y = Q_C' + Q_B' Q_A'$$

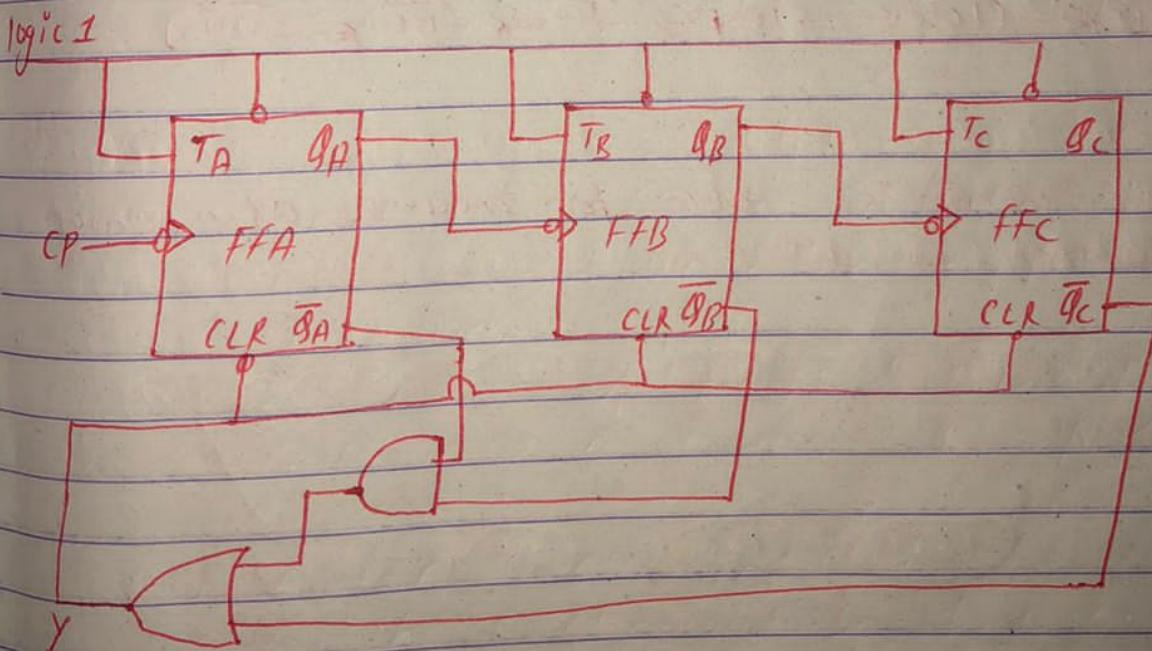


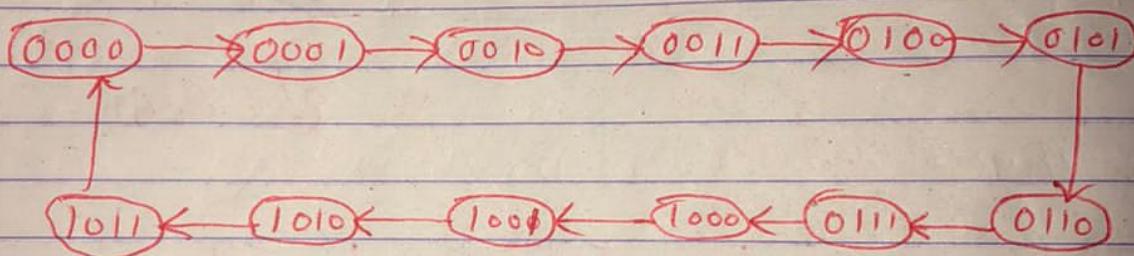
fig: Mod-5 ripple counter

Q1) Design a mod-12 asynchronous counter.

The number of states through which counter passes = 12
The states will go from 0 to 11.

To count 12 states, we need 4-bit ripple counter.
i.e Number of flipflop = 4

The state diagram for mod-12 asynchronous counter is given below:-

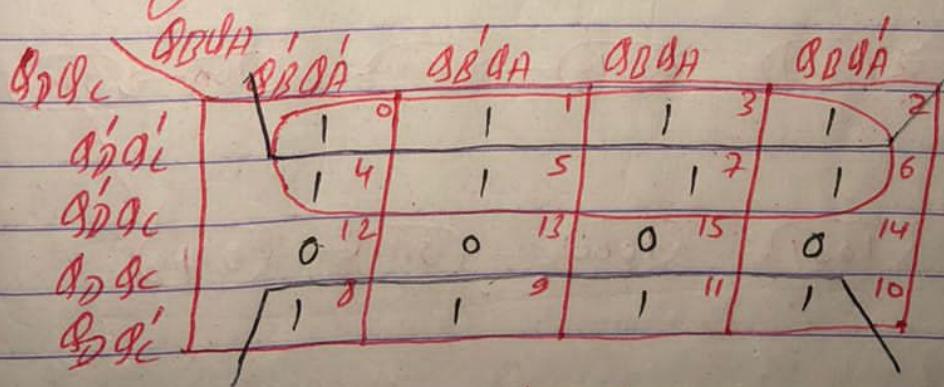


The transition table for mod-12 asynchronous counter is as:-

324

State	Q_D	Q_C	Q_B	Q_A	Reset logic (y)
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

Using k-map for simplification,



$$y = Q_D' + Q_C'$$

Logic 1

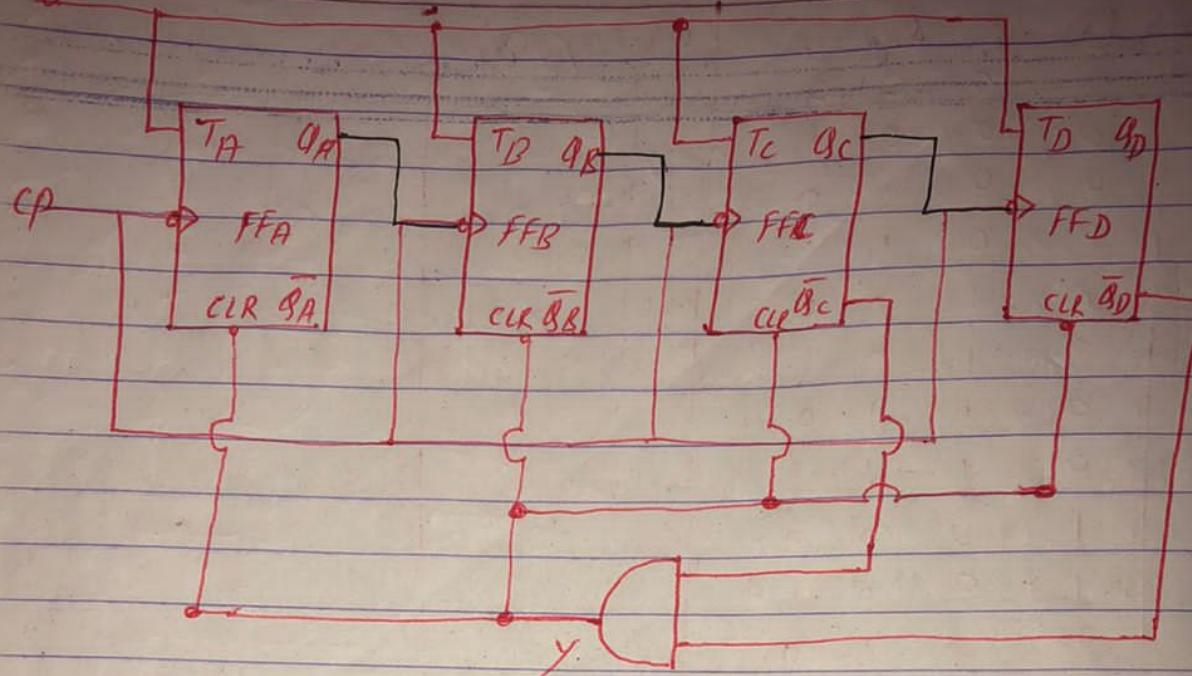
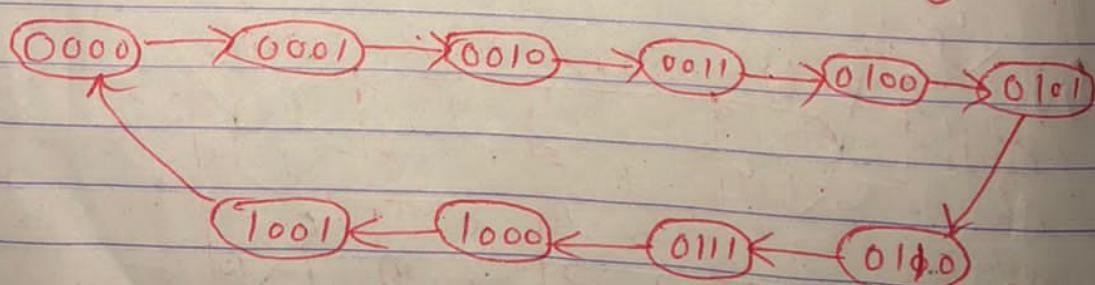


fig: logic diagram of mod-12 asynchronous counter

(a) Design a mod-10 (BCD counter) asynchronous counter.

Soln

The state diagram for BCD counter is given below:-



State	q_D	q_C	q_B	q_A	Reset logic (y)
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

~~Q~~

Using k-map for simplification,

$q_D q_C$	$q_B q_D$	$q_B' q_D'$	$q_B q_A$	$q_B' q_A'$
$q_D q_C$				
$q_D' q_C$	1 0	1 1	1 3	1 2
$q_D' q_C'$	1 4	1 5	1 7	1 6
$q_D q_C'$	0 12	0 13	0 15	0 14
$q_D q_C'$	0 8	1 9	0 11	0 10

$$y = q_D' + q_C' q_B'$$

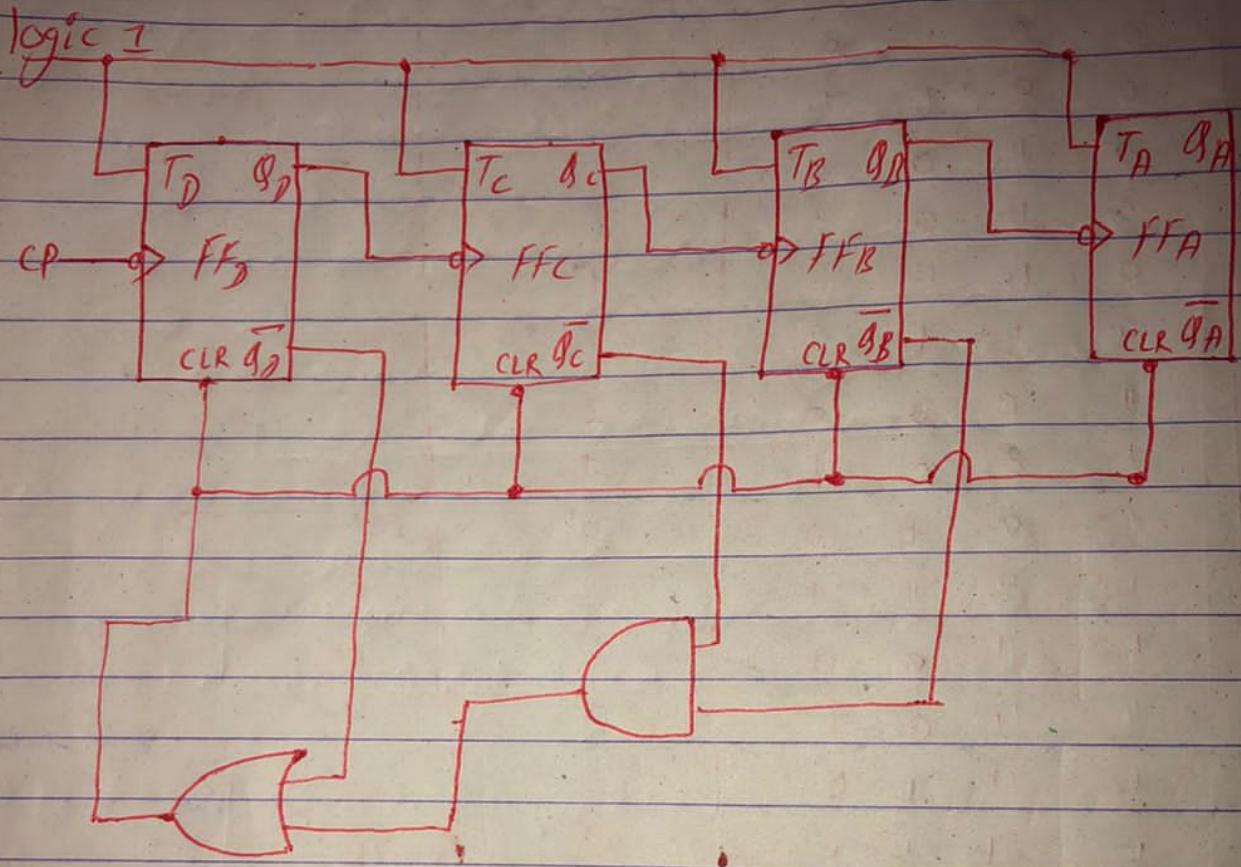


fig: mod-10 (BCD) Asynchronous counter

Design of Sequential Circuits

The design of a synchronous sequential circuit starts from a set of specifications and culminates (ends) in a logic diagram or a list of Boolean functions from which a logic diagram can be obtained.

The first step in the design of sequential circuits is to obtain a stable table or an equivalence representation, such as a state diagram.

A synchronous sequential circuit is made up of flip flops and combinational gates. The design of the circuit consists of choosing the flip-flops and then finding the combinational structure which, together with the flip-flops, produces a circuit that fulfills the required specifications.

The number of flip flops is determined from the number of states needed in the circuit.

The recommended steps for the design of sequential circuits are below:-

Specify the problem
(word description of the circuit)

Derive the State diagram

Obtain the State table

The number of states may be reduced
by State reduction method

Determine the number of flipflops
needed

Choose the type of flipflops to be
used

Derive excitation equations

Using the map or any other simplification
method, derive the output functions and
the flip-flop input functions

Draw the logic diagram

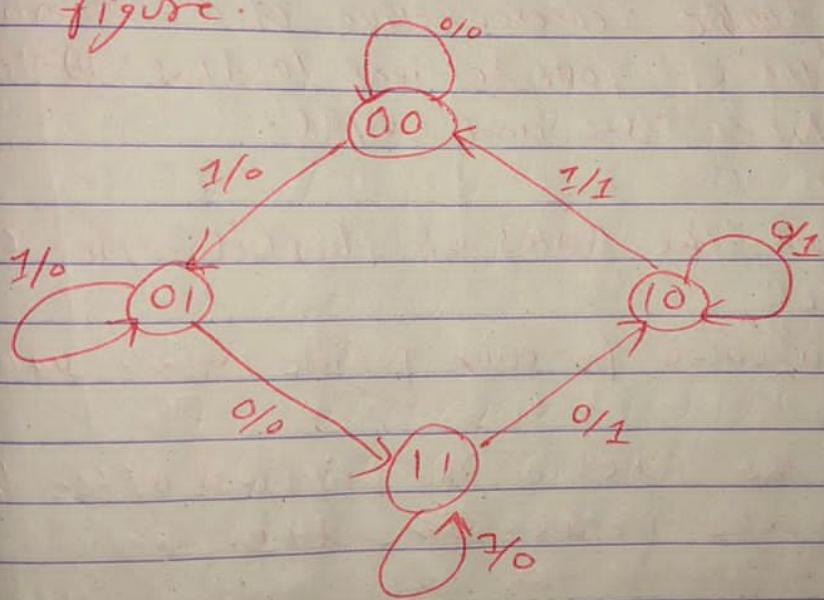
fig: Block diagram of design procedure of
sequential circuits.

State diagram

The information available in a state table can be represented graphically in a state diagram. In this diagram, a state is represented by a circle, and the transition between states is indicated by directed lines (arcs) connecting the circles.

A state diagram is a very convenient way to visualize the operation of a flip-flop or even of large sequential components.

An example of a state diagram is shown in figure.



The binary number inside each circle identifies the state of the flipflop. The directed lines are labelled with two binary numbers separated by slash (/) viz (input/output).

331

For example, the directed line from State 00 to 01 is labelled I/O, meaning that, if the sequential circuit is in a present state 00 and the input is 1, then the next state is 01 and the output is 0. If it is in a present state 00 and the input is 0, it will remain in the same state. A directed line connecting a circle with itself indicates that no change of state occurs.

State Table

The state table representation of a sequential circuit consists of four sections labelled present state, Input, Next state and output.

- 1) present state: Shows the states of flipflops before clock pulse.
- 2) Input: gives a ^{input} value for each possible present state
- 3) Next state: Shows the states of the flipflops after the clock pulse i.e at time $t+1$.
- 4) Output: lists the value of the output variables during the present state.

State Reduction

The reduction of the number of flip flops in a sequential circuit is referred to as the state reduction. State-reduction algorithms are concerned with procedures for reducing the number of states in a state table while keeping the external input-output requirements unchanged.

(a) Reduce the state table of a sequential circuit given.

present state	Next State		output	
	$n=0$	$n=1$	$n=0$	$n=1$
(A)	B	C	1	0
B	F	D	0	0
C	D	E	1	1
D	F	E	0	1
E	A	D	0	0
(F)	B	C	1	0

SOLN

From the table, we can see that the next state and output are same for both A and F. Hence, F can be replaced by A in the table. and now F is removed.

present state	next state		output	
	$n=0$	$n=1$	$n=0$	$n=1$
A	B	C	1	0
B	A	D	0	0
C	D	E	1	1
D	A	E	0	1
E	A	D	0	0

From table above, we can see that the next state and output is same for the both B and E.
So, Simple remove row-E from table and replace E by B.

present state	next state		output	
	$n=0$	$n=1$	$n=0$	$n=1$
A	B	C	1	0
B	A	D	0	0
C	D	B	1	1
D	A	B	0	1

Since there are no more similar next states and outputs, This is the final reduced state table.

The corresponding state diagram can be represented by following way:-

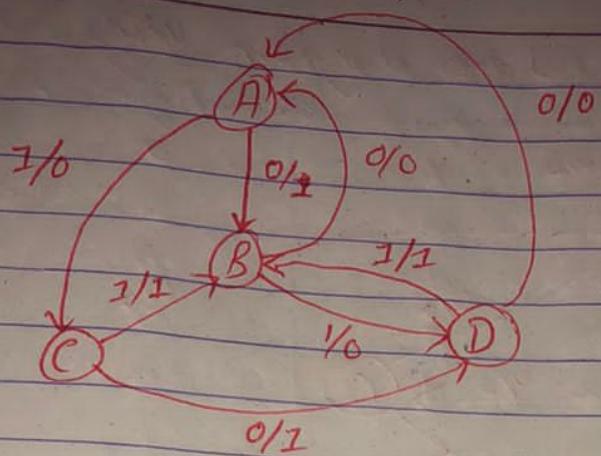


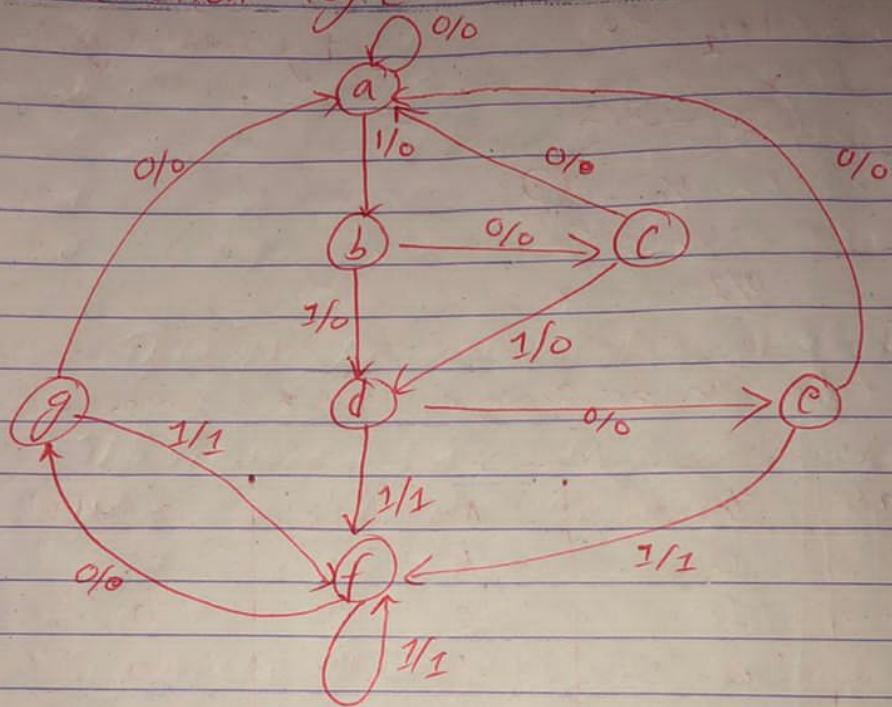
fig: state diagram

The state table can be arranged in other manner as:-

Present state	Input(X)	Next state	Output(Y)
A	0	B	1
A	1	C	0
B	0	A	0
B	1	D	0
C	0	D	1
C	1	B	1
D	0	A	0
D	1	B	1

335

(Q) Reduce the given state diagram using State reduction logic.

SOLN

The state table for the corresponding state diagram is drawn as:-

Present State	Next State		Output	
	$n=0$	$n=1$	$n=0$	$n=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	0
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

From the table, we can see that the next state and output are same for both e and g. So, we can remove 'g' from the table and replace g with e.

Present State	Next State		Output	
	$n=0$	$n=1$	$n=0$	$n=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

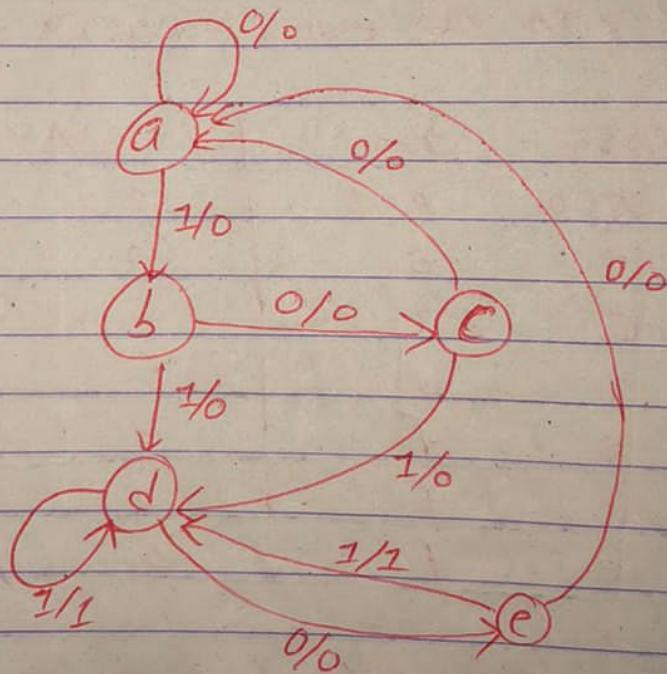
Again, from the state table, we can see that the next state and output are same for both d and f. So we can remove 'f' from table and replace f with d.

337

Present State	Next State		Output	
	$n=0$	$n=1$	$n=0$	$n=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1

This is the reduced state table.

The reduced state diagram can be drawn as:-



The Modified state table can be written as:-

Present State	Input (n)	Next State	Output
a	0	a	0
a	1	b	0
b	0	c	0
b	1	d	0
c	0	a	0
c	1	d	0
d	0	c	0
d	1	d	1
e	0	a	0
e	1	d	1

Q) Design a synchronous sequential circuit whose state diagram is shown in figure using J-K flip-flop.

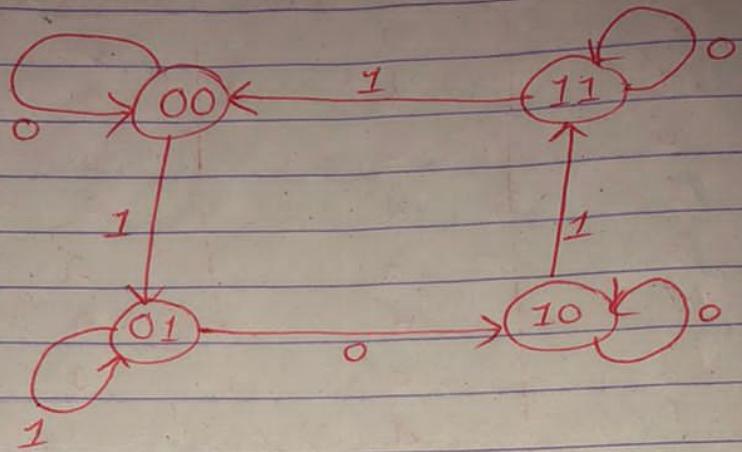


fig: State diagram

SOL

From the state diagram, the state table can be generated.

Present State		Next State	
A	B	$J = 0$	$J = 1$
0	0	0 0	0 1
0	1	1 0	0 1
1	0	1 0	1 1
1	1	2 2	0 0

The state table can be modified as:-

Present State	Input		Next State	
A B	n		A ⁺	B ⁺
0 0	0		0	0
0 0	1		0	1
0 1	0		1	0
0 1	1		0	1
1 0	0		1	0
1 0	1		1	1
1 1	0		1	1
1 1	1		0	0

Since, there are four states, two flipflops are needed i.e FFA and FFB.

The excitation table for JK flipflop is

\emptyset	\emptyset^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Let us find out the inputs of J-K flipflop.

341

Present State Input			Next State		Flipflop Inputs	
A	B	n	A'	B'	J _A	K _A
0	0	0	0	0	0	X
0	0	1	0	1	0	X
0	1	0	1	0	1	X
0	1	1	0	1	0	X
1	0	0	1	0	X	0
1	0	1	1	1	X	0
1	1	0	1	1	X	0
1	1	1	0	0	X	1

Using k-map simplification we find out the input for the flipflop.

For J_A

A	B'n	B'n'	B'n	Bn	Bn'
A'	0 ⁰	0 ¹	0 ³	1 ²	
A	X ⁴	X ⁵	X ⁷	X ⁶	

For K_A

A	B'n	B'n'	B'n	Bn	Bn'
A'	X ⁰	X ¹	X ³	X ²	
A	0 ⁴	0 ⁵	1 ⁷	0 ⁶	

$$\therefore J_A = B'n'$$

$$\therefore K_A = Bn$$

For J_B

A	B'n	B'n'	B'n	Bn	Bn'
A'	0 ⁰	1 ¹	X ³	X ²	
A	0 ⁴	1 ⁵	X ⁷	X ⁶	

For K_B

A	B'n	B'n'	B'n	Bn	Bn'
A'	X ⁰	X ¹	0 ³	1 ²	
A	X ⁴	X ⁵	1 ⁷	0 ⁶	

$$\therefore J_B = n$$

$$\begin{aligned} K_B &= A'n' + An \\ &= A \oplus n \quad (\text{X-NOR}) \end{aligned}$$

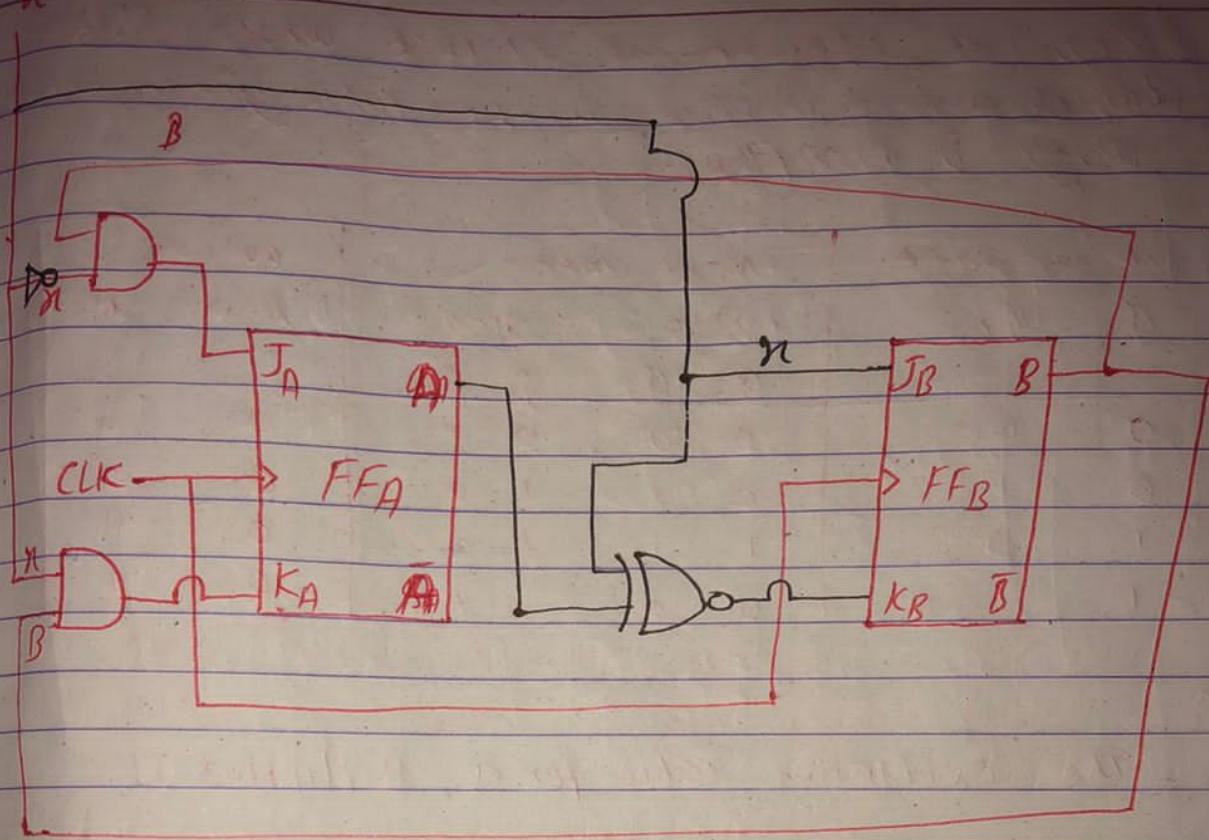


fig: Logic diagram of the sequential circuit.

343

Q) Design a sequential circuit whose state tables are specified in the table given below, using D-flipflops.

present state	next state		output	
$q_0 \ q_1$	$n=0$	$n=1$	$n=0$	$n=1$
0 0	0 0	0 1	0	0
0 1	0 0	1 0	0	0
1 0	1 1	1 0	0	0
1 1	0 0	0 1	0	1

SOLN

The excitation table for a D-flipflop is:-

Q	Q^+	D
0	0	0
0	1	1
1	0	0
1	1	1

To derive the excitation table for the circuit design which is shown in table using D-flipflop is given as:-

Present State	Input	Next State	Flipflop inputs	Output
q_0	q_1	q_0^+	D_0	Z
0	0	0	0 0	0
0	0	0	0 1	0
0	1	0	0 0	0
0	1	1	1 0	0
1	0	1	1 1	0
1	0	1	1 0	0
1	1	0	0 0	0
1	1	0	0 1	1

Using k-map simplification, let us find out the Boolean expression for D_0 , D_1 , and Z .

For D_0

$q_1 n$	$q_1' n'$	$q_1' n$	$q_1 n'$	$q_1 n$
q_0'	0 0	0 1	1 3	0 2
q_0	1 4	1 5	0 7	0 6

$$D_0 = q_0 q_1' + q_0' q_1 n$$

For D_1

$q_1 n$	$q_1' n'$	$q_1' n$	$q_1 n'$	$q_1 n$
q_0'	0 0	0 1	1 3	0 2
q_0	1 4	0 5	1 7	0 6

$$D_1 = q_0' q_1' n + q_0 q_1' n' + q_0 q_1 n$$

for Z ,

$q_1 n$	$q_1' n'$	$q_1' n$	$q_1 n'$	$q_1 n$
q_0'	0 0	0 1	0 3	0 2
q_0	0 4	0 5	1 7	0 6

$$Z = q_0 q_1 n$$

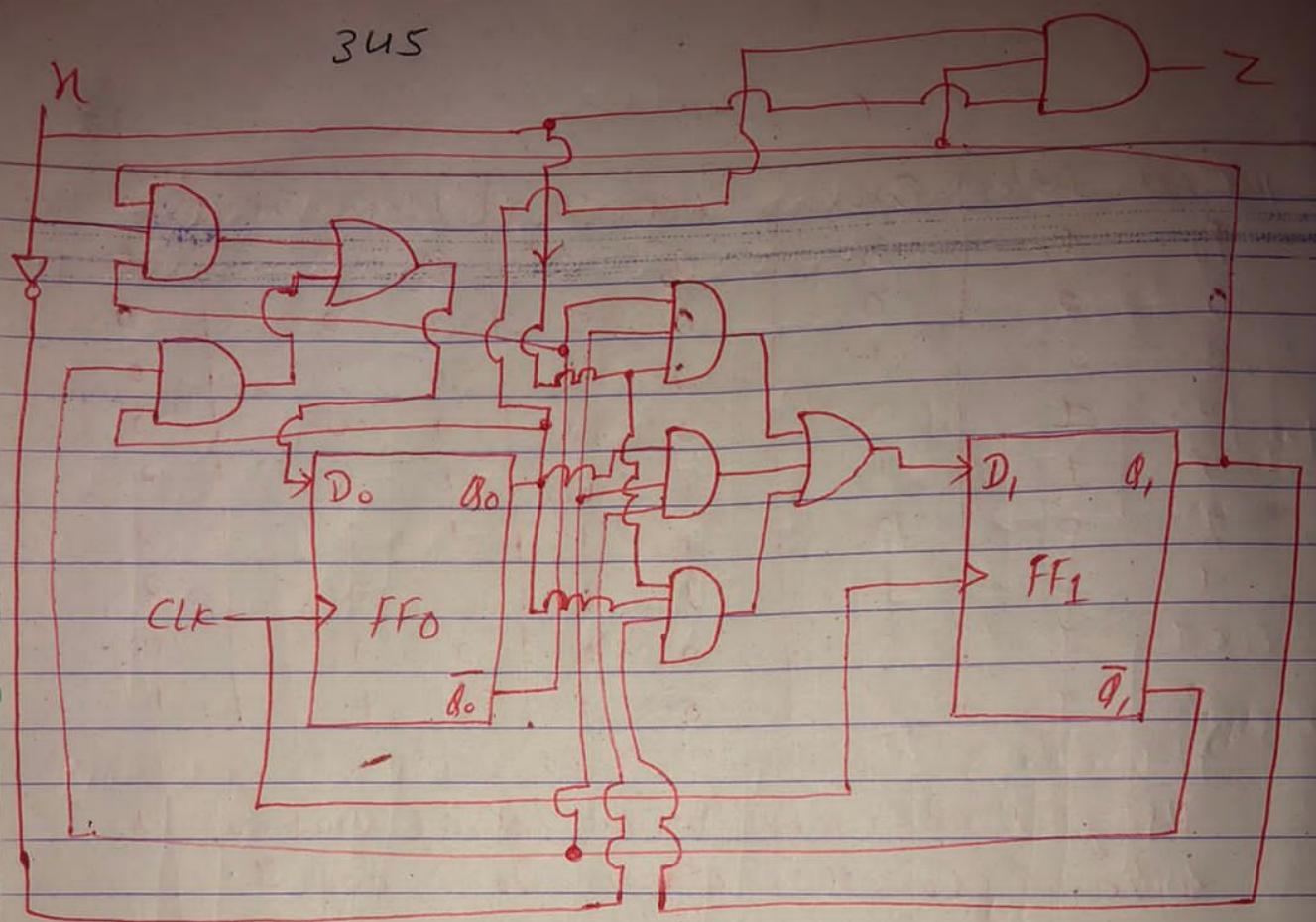


fig. Logic diagram of the sequential circuit.

Cascaded Counters

For counting the higher number of modules (States) two or more counter are connected in a cascading form which is known as Cascaded Counter. A cascaded counter is constructed by connecting the output of first counter to the input to the second counter, the output of second counter is connected as a input of ~~3rd~~ third counter.

Figure below illustrate MOD-M.N.P counter connected in a cascaded form and count up to $M \times N \times P - 1$ i.e total state $M \times N \times P$.

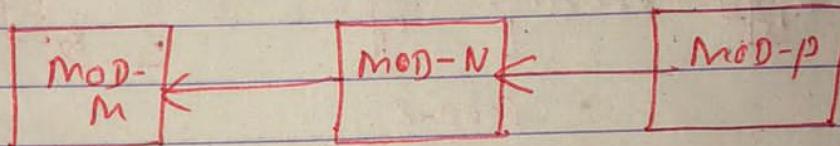


fig: MOD-M×N×P counter.

Example: MOD-640 Asynchronous counter

Figure below illustrate the MOD-640 counter connecting in a cascaded form and count total state $8 \times 8 \times 10$. The clock pulse is provided to the first flip flop. The output of first flip flop is converted to the input of second flip flop and the output of second flip flop is connected to the clock of third flip flop. It is continued till 9th flip flop. In the

347

figure, two MOD-8 counter and a MOD-10 counter
are connected to each other.

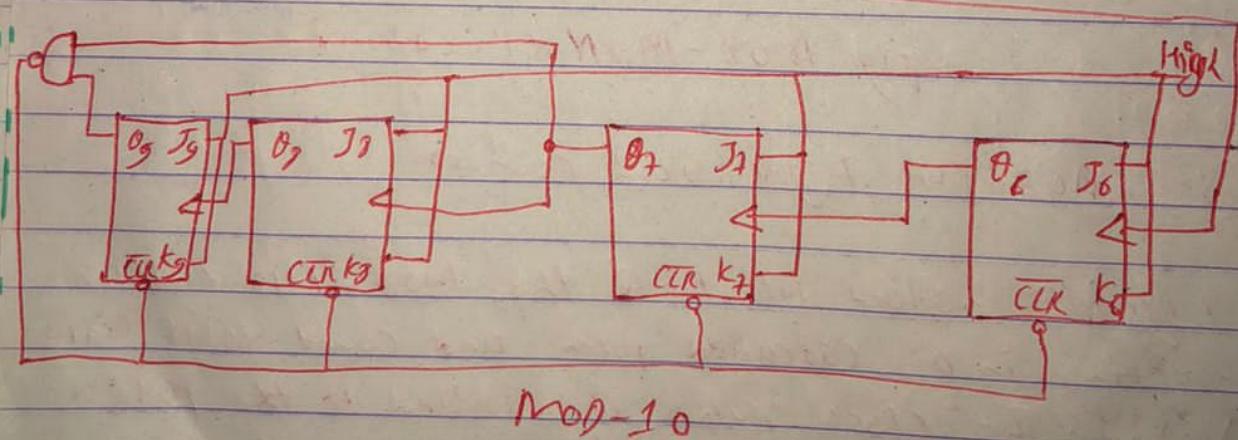
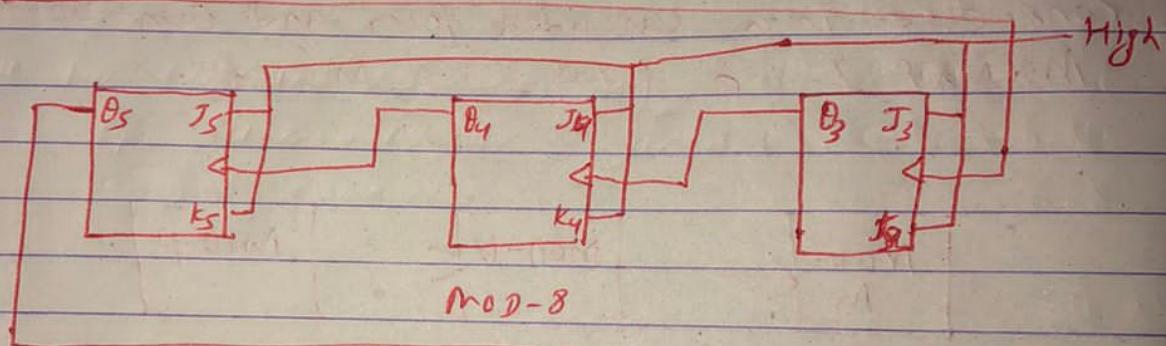
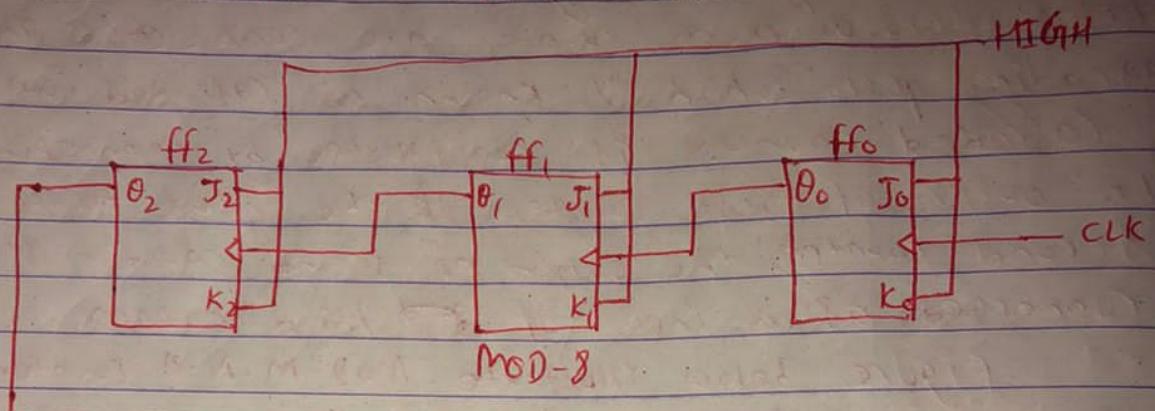


fig: MOD-640 Asynchronous counter

Example 2: MOD-156 Asynchronous counter.

Figure below illustrate the MOD-156 counter connecting in a cascade form and count total state 12×13 . The clock pulse is provided to the first flip flop only. The output of 1st flip flop is connected to the 2nd flip flop, the output of 2nd is connected to the CLR of 3rd flip flop and so on.

MOD-12 is shown in the upper part whereas MOD-13 is shown in the lower part of figure. After the ff₃, NAND gate is used and clear has provided the value to all the flipflops. The output of ff₃ is connected as a clock to the ff₄ of MOD-13 counter.

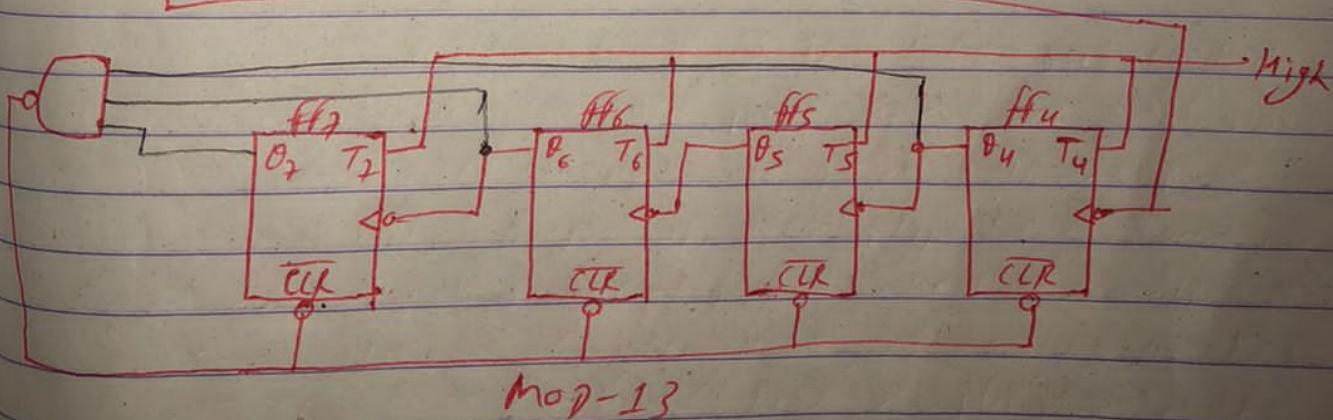
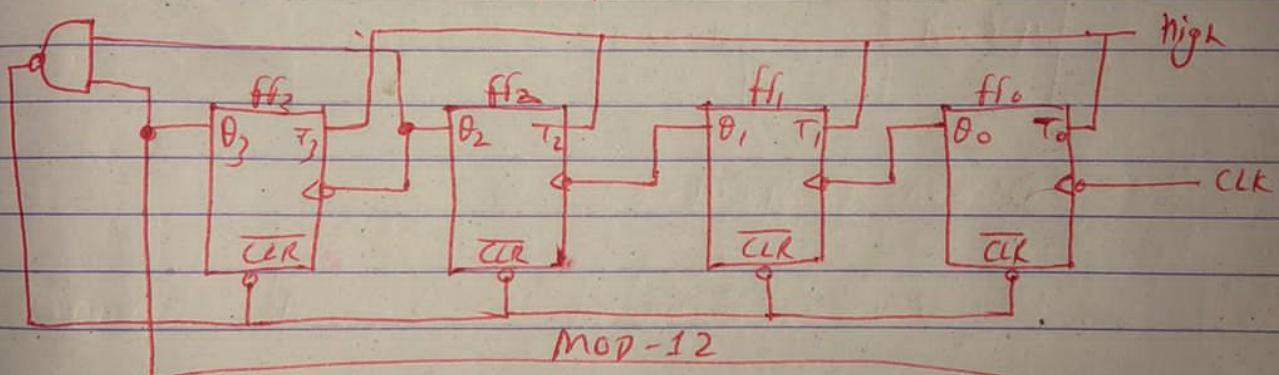


fig: MOD-156 Asynchronous Counter

Applications/Uses of Counters

- 1) Digital clocks
- 2) Bank Token counters at cashier section
- 3) Counting system on automated production line
- 4) Digital stop watches
- 5) Timers for automated water supply
- 6) Sensor controlled counters
- 7) Counting general objects in any system.

Note: For cascaded counter, simply take $M \times N$. i.e.
if 64-mod counter is given, we can use
 8×8 counter.